**UNIVERSIDAD NACIONAL DE SAN AGUSTIN**

**Esc. Ciencia de la Computación**

**Trabajo de Algoritmos Paralelos**

**Alumno:** Diego Ranilla Gallegos

1. **Implement in C the simple three-nested-loop version of the matrix product and try to evaluate its performance for a relatively large matrix size.**

void multiplicacion(int A[tam][tam],int B[tam][tam],int C[tam][tam]){

for(int i=0;i<tam;i++){

for(int j=0;j<tam;j++){

for(int k=0;k<tam;k++){

C[i][j] += A[i][k]\*B[k][j];

}

}

}

}

Prueba:

Cantidad de datos = matriz A y B de 500 x 500

Tiempo de ejecucion = 0.599149 segundos

1. **Implement the blocked version with six nested loops to check whether you can observe a significant gain.**

void multiplicacion2(int A[tam][tam],int B[tam][tam],int C[tam][tam]){

for(int i1=0;i1<tam;i1=i1+bloque){

for(int j1=0;j1<tam;j1=j1+bloque){

for(int k1=0;k1<tam;k1=k1+bloque){

for(int i=i1;i<=min(i1+bloque-1,tam);i++){

for(int j=j1;j<=min(j1+bloque-1,tam);j++){

for(int k=k1;k<=min(k1+bloque-1,tam);k++){

C[i][j] += A[i][k]\*B[k][j];

}

}

}

}

}

}

}

Prueba:

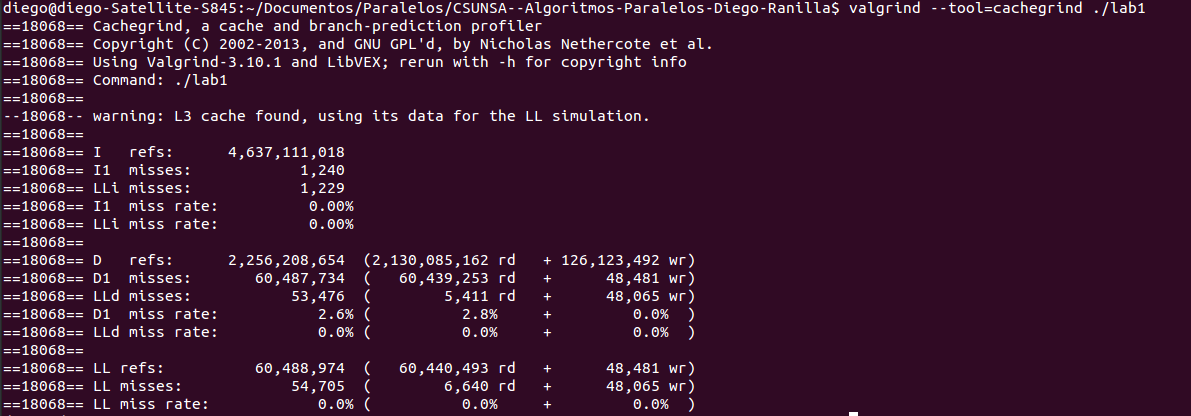
Cantidad de datos = matriz A y B de 500 x 500

Tiempo de ejecucion = 1.09454 segundos

1. **Execute these algorithms step by step to get a good understanding of data movements between the cache and the memory and try to evaluate their respective complexity in term of distant memory access.**

1. **Execute these two versions of the code with valgring and kcachegrind to get a precise evaluation of their performance in term of cache misses.**

**Algoritmo I:**

****

**Algoritmo II:**

